# A METHOD OF, AND SYSTEM FOR, REPLACING EXTERNAL LINKS IN ELECTRONIC DOCUMENTS

The present invention relates to a method of, and system for, replacing external links in electronic documents such as email with links which can be controlled. One use of this is to ensure that email that attempts to bypass email content scanners no longer succeeds. Another use is to reduce the effectiveness of web bugs.

Content scanning can be carried out at a number of places in the passage of electronic documents from one system to another. Taking email as an example, it may be carried out by software operated by the user, e.g. incorporated in or an adjunct to, his email client, and it may be carried out on a mail server to which the user connects, over a LAN or WAN, in order to retrieve email. Also, Internet Service Providers (ISPs) can carry out content scanning as a value-added service on behalf of customers who, for example, then retrieve their content-scanned email via a POP3 account or similar.

One trick which can be used to bypass email content scanners is to create an email which just contains a link (such as an HTML hyperlink) to the undesirable or "nasty" content. Such content may include viruses and other varieties of malware as well as potentially offensive material such as pornographic images and text, and other material to which the email recipient may not wish to be subjected, such as spam. The content scanner sees only the link, which is not suspicious, and the email is let through. However, when viewed in the email client, the object referred to may either be bought in automatically by the email client, or when the reader clicks on the link. Thus, the nasty object ends up on the user's desktop, without ever passing through the email content scanner.

It is possible for the content scanner to download the object by following the link itself. It can then scan the object. However, this method is not foolproof – for instance, the server delivering the object to the content scanner may be able to detect that the request is from a content scanner and not from the end user. It may then serve up a different, innocent object to be scanned. However, when the end-user requests the object, they get the nasty one.

The present invention seeks to reduce or eliminate the problems of embedded links in electronic documents and does so by having the content scanner attempt to follow a link found in an electronic document and scan the object which is the target of the link. If the object is found to be acceptable from the point of view of content-scanning criteria, it is retrieved by the scanner and stored on a local, trusted server which is under

the control of the person or organisation operating the invention. The link in the electronic document is adjusted to point at the copy of the object stored on the trusted server rather than the original; the document can then be delivered to the recipient without the possibility that the version received by the recipient differs from the one originally scanned. Note that it does not matter to the principle of the invention whether the linked object is stored on the trusted server before or after it has been scanned for acceptability; if it is stored first and found unacceptable on scanning, the link to it can simply be deleted.

If the object is not found to be acceptable, one or more remedial actions may be taken: for example, the link may be replaced by a non-functional link and/or a notice that the original link has been removed and why; another possibility is that the electronic document can be quarantined and an email or alert generated and sent to the intended recipient advising him that this has been done and perhaps including a link via which he can retrieve it nevertheless or delete it. The process of following links, scanning the linked object and replacing it or not with an embedded copy and an adjusted link may be applied recursively. An upper limit may be placed on the number of recursion levels, to stop the system getting stuck in an infinite loop (e.g. because there are circular links) and to effectively limit the amount of time the processing will take.

Thus according to the present invention there is provided a content scanning system for electronic documents such as emails comprising:

a) a link analyser for identifying hyperlinks in document content;

b) means for causing a content scanner to scan objects referenced by links identified by the link analyser and to determine their acceptability according to predefined rules, the means being operative, when the link is to an object external to the document and is determined by the content analyser to be acceptable, to retrieve the external object and modify the document by replacing the link to the external object by one to a copy of the object stored on a trusted server.

The invention also provides a method of content-scanning electronic documents such as emails comprising:

a) using a link analyser for identifying hyperlinks in document content;

b) using a content scanner to scan objects referenced by links identified by the link analyser and to determine their acceptability according to predefined rules, the means being operative, when the link is to an object external to the document and is determined by the content analyser to be acceptable, to retrieve the external object and

modify the document by replacing the link to the external object by one to a copy of the object stored on a trusted server.

Thus the content scanner can follow the link, and download and scan the object. If the object is judged satisfactory, a copy of it is stored on the trusted server, and the link to the external object replaced by a link to that copy.

One trick used by spammers is to embody 'web bugs' in their spam emails. These are unique or semi-unique links to web sites – so a spammer sending out 1000 emails would use 1000 different links. When the email is read, a connection is made to the web site, and by finding which link has been hit, the spammer can match it with their records to tell which person has read the spam email. This then confirms that the email address is a genuine one. The spammer can continue to send email to that address, or perhaps even sell the address on to other spammers.

By following every external link in every email that passes through the content scanner, all the web bugs the spammer sends out will be activated. Their effectiveness therefore becomes much reduced, because they can no longer be used to tell which email addresses were valid or not.

The invention will be further described by way of non-limiting example with reference to the accompanying drawings, in which:-

Figure 1 shows the "before" and "after" states of an email processed by an embodiment of the present invention; and

Figure 2 shows the email processor of a system embodying the present invention; and

Figure 3 shows an object server for providing objects referenced by links in email which has been rebuilt by the system of Figure 2.

Figure 1a shows an email 1 formatted according to an internet (e.g. SMTP/MIME) format. The body includes a hypertext link 2 which points to an object 3 on a web server 4 somewhere on the internet. The object 3 may for example be a graphical image embedded in a web page (e.g. HTML or XML).

Figure 1b shows the situation after the email 1 has been processed by the system to be described below and the content pointed at by the link 2 has been judged to be acceptable. The content, i.e. image 3 has been copied to an object server 5 as image 3'; the object server 5 is hosted on a secure server machine 6 (or array of such machines) under the control of the person or organisation operating the system (e.g. an ISP). The original link 2 has been replaced by a new link 2' pointing at the image 3' stored on the secure or

trusted server 6. The server 6 operates in the security domain of the operator of the system and has access permissions associated with the stored content objects such as 3' which enable eventual recipients of emails such as 1, or more strictly speaking their email client software to follow the link 2' and retrieve the linked-to object. Of course, the access permissions of server 6 should prevent persons or software without appropriate security credentials from writing to the linked-to object storage area.

Figures 2 and 3 illustrate a system according to the present invention. Although the invention is not limited to this application, this example embodiment is given in terms of a content scanner operated by an ISP to process email stream e.g. passing through an email gateway.

Figure 2 shows the part of the system which processes emails and modifies them to replace links to objects on untrusted servers such as 4 by links to objects on trusted server 6, where the linked-to object is considered to be acceptable content. Figure 3 shows the object server which provides the linked-to objects when recipients follow the processed links in their emails.

The part 100 of the system which is shown in Figure 2 may operate as follows in respect of each item of email delivered to an input 101.

1. The email is analysed by analyser 102 to determine whether it contains external links. This determination may be made, for example, by scanning it for standard markup tags which point to external content or objects, for example the <A> and <IMAGE> tags in HTML. If none are found, steps 2 to 5 are omitted and the email is delivered unprocessed to output 103 via path 104.

2. The analyser 102 operates in concert with a link replacer 105 to process links to external objects. For each link, the link replacer 105 creates a new link which is stored in a link database 106. The new link is generated by a process guaranteed to generate unique links each time. A database entry is created, tying the original link and the new, replacement, one together.

3. An email rebuilder 107 rebuilds the email with each link replaced by its new counterpart stored in link database 106 and the rebuilt email is forwarded on.

4. When the email is read, the part 200 of the system illustrated in Figure 3 comes into play. The new link may be requested either by the email client software, or by the person reading the email clicking or otherwise selecting the link. This generates a request retrieve an object from the trusted server 6. The server 6 looks up in the link

database 106 to find the original object, and retrieves it. If it cannot be retrieved, go to step 8.

5. The external objects are scanned for pornography, viruses, spam and other undesirables. If any are found, go to step 8.

6. The external objects are analysed to see whether they contain external links. If the nesting limit has been reached, go to step 8. Otherwise each external link is replaced by a new link in a manner similar to step 2, and a database entry is created, tying the old and new links together.

7. The rebuilt object is forwarded to the requester, and the process ends

8. If processing arrives here, an undesirable object has been found, or the object could not be retrieved, or the nesting limit has been reached. The system can now take some appropriate error action, such as returning an error message, alerting an operator or returning a default object.

Figure 3 shows the object server 300 which services requests received at an input 301 to retrieve a linked-to object on an entrusted server, scan it for acceptability and, if acceptable, to store it on the secure "safelink" server 6. An object locator 300 locates the linked-to object, e.g. on the internet and initiates a retrieval operation by which the object is retrieved by the retriever 303. This retrieval process takes place using the internet protocol appropriate to the link and linked-to object. If the retrieval fails, an error handler 304 is invoked. If successful, the object is processed by an object control scanner which makes a determination of whether the content is acceptable. If it is not, the error handler 304 is invoked, otherwise an object returner 306 returns the object and stores it on the trusted server 6.

Example

The following email contains a link to a website.

```
Subject: email with link
Subject:
Date: Thu, 9 May 2002 16:17:01 +0600
MIME-Version: 1.0
Content-Type: text/html;
Content-Transfer-Encoding: 7bit

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
</HEAD>
<BODY bgColor=3D#ffffff>
<DIV> </DIV>
This is some text<BR>
```

```
<DIV><IMAGE src="http://www.messagelabs.com/threatlist"
</DIV>
This is some more text<BR>
</BODY></HTML>
```

A new link is generated: http://safelink.com/09052002161710a33071ef407.gif, the email is updated, and a database entry is generated.

Database Entry

Old link: http://www.messagelabs.com/threatlist.gif
New link: http://safelink.com/09052002161710a33071ef407.gif

Updated email

```
Subject: email with link
Subject:
Date: Thu, 9 May 2002 16:17:01 +0600
MIME-Version: 1.0
Content-Type: text/html;
Content-Transfer-Encoding: 7bit

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
</HEAD>
<BODY bgColor=3D#ffffff>
<DIV> </DIV>
This is some text<BR>
<DIV><IMAGE src="http://safelink.com/09052002161710a33071ef407.gif"
</DIV>
This is some more text<BR>
</BODY></HTML>
```

When the email is read, the email client may try and download the image referred to by the link. However, it will try and retrieve the image from http://safelink.com/09052002161710a33071ef407.gif rather than http://www.messagelabs.com/threatlist.gif

The safelink.com server will lookup http://safelink.com/09052002161710a33071ef407.gif in the database, and find that the original link is http://www.messagelabs.com/threatlist.gif. It will download the object, and scan it, perhaps for pornography or other inappropriate content.

If the scan shows the object is harmless, it can be passed back to the original requestor.

**Example Link Generator**

The link can be generated by processing the name of the server generating the link, the current time, the process id, a number that increments each time and random number. This will all be appended to an appropriate reference to the 'safelink' server.

Thus if the server generating the link is mail2071.messagelabs.com, the reference is for the http protocol the time is 27 Jan 2003, 17:45:01 the process id is 1717, 27 references have already been generated and the safelink server is safelink.com, then a typical link might be:

http://safelink.com/mail2071_messagelabs_com/27012003174501/1717/28/10131354834

**Other Improvements**

Other improvements can be added to the system. For instance, the system does not have to wait until the object is first requested. It may proactively fetch the object ahead of time, scan it, and either cache the object or remember that the scan did not pass. This will cut down on delays when the object is requested. If all links are followed then this will activate all web bugs placed in the email, thereby much reducing their effectiveness.

The system can also cache and intelligently remember which objects have been retrieved – if five emails contain the same original link, then even though they will end up with five different new links the referred to object only needs to be retrieved once and not five times.

The system might want to ensure the same filename extension is used for the old and new links.